Point-to-multipoint connectivity and recovery in Ethernet Label Switching

Wouter Tavernier[1], Dimitri Papadimitriou[2], Didier Colle[1], Mario Pickavet[1], and Piet Demeester[1]

[1]Ghent University, IBCN-IBBT, INTEC, Gaston Crommenlaan 8 bus 201, B-9050 Gent, Belgium
[2]Alcatel-Lucent Bell, Copernicuslaan 50, B-2018 Antwerpen, Belgium

Abstract

Carrier-grade Ethernet is a recent technology evolution where the most attractive features of bridged Ethernet are enhanced with highly capable control and forwarding features. High speed physical (PHY) interfaces at low cost, combined with transport network capabilities make it the choice of the future for many network operators. We show that Ethernet Label Switching (ELS) is capable of efficiently setting up and recovering both point-to-point and point-to-multipoint data paths. We have evaluated both techniques in an emulation setup and show that high speed recovery is possible in a realistic setting.

1 Introduction

For decades Ethernet is dominating the LAN environment. Ethernet bridging has become a synonym for a cheap, plug-and-play and highly compatible network technology. The common Ethernet usage in enterprise networks together with ubiquitous deployment trends results in an ever increasing demand to providers for Ethernet services enabling to interconnect several branches of companies (e.g., Virtual Private LAN Service (VPLS) and Virtual Private Wire Services (VPWS).

This shift towards packet-oriented inter-connection services has also its consequences on the transport technology that is being envisioned by operators. Why would they still use more expensive circuit-based optical equipment, involving several conversion layers, e.g., GFP, VCAT, etc., if the majority of services is becoming more and more packet-based (elastic and streaming traffic). This reasoning will become even more striking, given the highly increasing Ethernet PHY speeds, going from 10 Gbps towards 40 and 100 Gbps. Therefore using Ethernet directly as a transport technology in access-, (metro-)aggregation or even core networks becomes more and more attractive.

However, base principles of Ethernet bridging of learning and flooding within a restricted virtual tree topology (based on Rapid Spanning Tree Protocol (RSTP)) is far too restrictive for use as a transport technology. This resulted in a spectrum of new Ethernet technology designs that were developed by IEEE, ITU-T and IETF. In this paper, we focus on Ethernet Label Switching (ELS) as one of the most attractive carrier-grade Ethernet technologies.

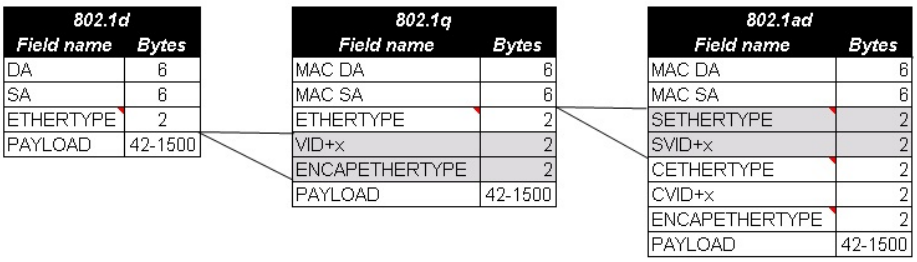


Figure 1: IEEE Ethernet framing standards

The structure of the paper is as follows. Section 2 describes the architecture and main functional blocks of ELS. An overview is given of possible recovery techniques for ELS point-to-point connectivity in Section 3. The next section (Section 4) describes how ELS is able to provide multipoint connectivity and corresponding recoverability. A realistic experimental setup is described in Section 5, where we show that a mix of point-to-point (P2P) and point-to-multipoint (P2MP) label switched paths (LSPs) can be efficiently combined and recovered in an emulation environment. Finally, a conclusion is given in Section 6.

2 Ethernet Label Switching

ELS is a connection-oriented forwarding scheme that is able to use Generalized Multi-Protocol Label Switching (GMPLS) as its control suite with Open Shortest Path First-Traffic Engineering (OSPF-TE) for routing and Resource reSerVation Protocol-Traffic Engineering (RSVP-TE) for signaling logical data paths over an Ethernet Network. ELS relies on the Provider Bridges (IEEE 802.1ad) recommendation to perform label switching in a similar way as performed in Multi-Protocol Label Switching (MPLS) as specified in RFC 3031. It encodes the label in the S-VLAN ID (S-VID) tag field of the related frame header (see Figure 1). The Ethernet S-VID label space has link local scope and local significance: thus providing 4096 (12 bits) values per interface and allowing intermediate ELS switches to translate the S-VID value resulting logically into a label swapping operation as it is the case in MPLS networks.

The logical data paths established using ELS are called Ethernet label switched paths (E-LSP). Intermediate nodes are called Ethernet Label Switching Router (E-LSR). Ingress/egress E-LSR where a LSP starts and ends, provide for a Ethernet Label Edge router (E-LER) functionality. Figure 2 describes the label operations along an Ethernet LSP.

When a native Ethernet frame arrives to the ingress LSR, its E-LER function based on the information of the frame header, pushes the corresponding label (i.e. adding an S-TAG with the appropriate S-VID value). Then, the Ethernet VLAN-labeled frame is forwarded along the Ethernet LSP. For each E-LSR, the label is swapped (i.e. that the incoming S-VID is translated into an outgoing S-VID as defined in IEEE 802.1ad). When the frame reaches the egress LSR, its E-LER function pops the label (the S-TAG and so the S-VID are removed). Finally, the frame is sent to its destination as a native Ethernet frame.

It is important to underline that ELS maintains a control state per data path but keeps the forwarding paradigm of existing Ethernet switches unchanged except for

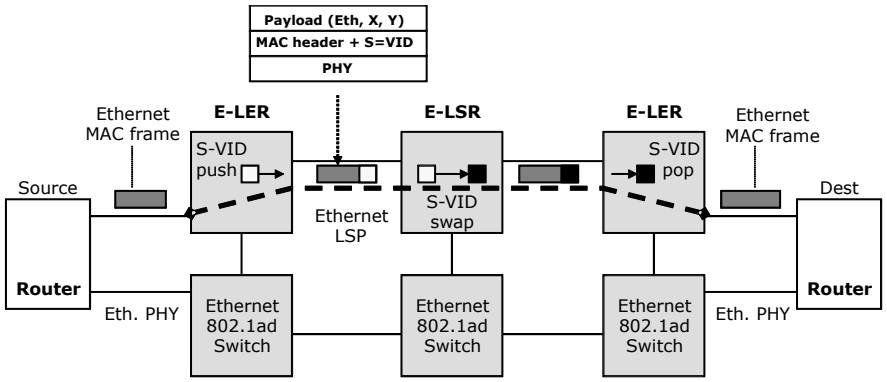


Figure 2: GELS LSP

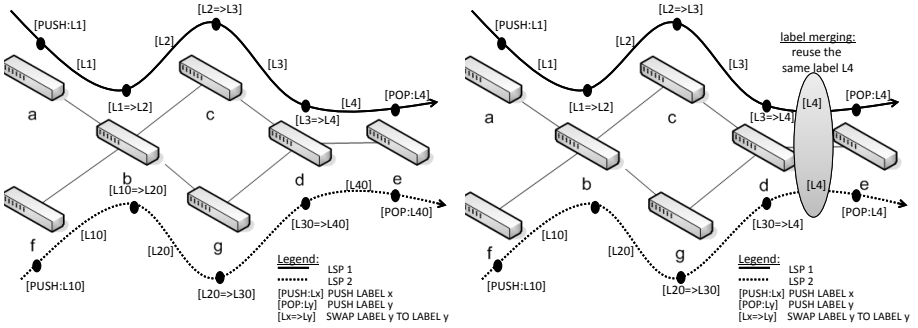
the fact that forwarding entries are defined per port. In a sense, signaling is used to restrict the incoming and outgoing S-VID per port. The rest of the forwarding process is performed as specified in IEEE 802.1ad.

2.1 Forwarding tables and merging capability

The logical forwarding behavior borrows from existing terminology of MPLS forwarding as defined in RFC 3031. In practice, this means that three main tables were developed such as to allow the setup of end-to-end ELS LSPs. Instead of using IP prefixes to match incoming frames in the head-end ELS switch to the configured ELS LSP, the incoming port index, possibly in combination with the incoming (customer) VLAN-tag, is determining the ELS LSP to be used, i.e., outgoing frames are tagged with the S-VID associated to that LSP on the corresponding outgoing link). The following tables and associated actions were implemented :

- *FTN (FEC-TO-NHLFE)*: The FTN table is used at the ingress (or head-end) ELS switch to match or classify incoming frames (based on their incoming interface and VLAN-tag) to entries in the NHLFE table.
- *ILM (Incoming Label Map)*: The ILM table is used at intermediate and egress (or tail-end) ELS switches to match or classify incoming ELS frames based on their incoming interface and S-VID label to entries in the NHLFE table. Together with this matching operation, a POP action is performed on the incoming S-VID label, meaning that the S-VID label is taken away from the frame header, such as to make PUSH actions subsequently possible.
- *NHLFE (Next Hop Label Forwarding Entry)*: The NHLFE table is used at ingress, intermediate and egress ELS switches in order to forward them into the ELS network having a S-VID label on which can be further switched. This implies that a PUSH action, meaning that a S-VID label is attached to the frame header.

The link-local significance and the above forwarding table characteristics allow an efficient label re-usage in the network. Figure 5 illustrates in a given network that two separate LSPs, sharing a common segment towards a destination, can use the same label. In this figure, the upper LSP (solid line for LSP 1) coincides with the lower LSP (dashed line for LSP 2) on the last segment of their path. If no label merging would be



(a) without merging

(b) with merging

Figure 3: Efficient label re-usage through merging

allowed, as in Figure 5 (a), LSP 1 and LSP 2 would need to allocate a different label for the link between node d and node e (L4 for LSP 1 and L40 for LSP 2). If merging is allowed, as in ELS, both LSP's can use the same label on the link d-e (L4 on the figure). On the lower level forwarding structure, label merging is allowed by coupling a two distinct ILM-entries (on the figure: frame with L3 from node c and frame with L30 from node d) to the same NHLFE entry (related to pushing label L40 and forwarding on link d-e in the figure).

3 Recovery of point-to-point (P2P) connectivity

The forwarding machinery of ELS allows for a wide range of recovery techniques point-to-point (p2p) data oaths (LSPs with one source and one destination), including both protection (recovery path is pre-computed and installed in the network) and dynamic re-routing a.k.a restoration (recovery path is computed after failure occurrence).

An overview of allowed recovery techniques is given in Figure 4. Link and node recovery can be obtained by using a detour sub-path for the corresponding link (link bypass, e.g., one can recover from failing link b-c by passing via node f) or node (node bypass, e.g., failing node b can be circumvented by passing via node g) upon failure occurrence. Extending the concepts of node and link bypass towards bypassing an arbitrary segment of an LSP is referred to as segment recovery (e.g, recovery from a failure between node a and node d can be done via a backup segment via node i and j in the figure). At last, recovery from an arbitrary failure along the path of an LSP, can be achieved using end-to-end recovery (e.g., using the backup path via node k and l). The IETF has defined a set of signaling extensions for GMPLS segment recovery using RSVP-TE in RFC 4873 and for end-to-end recovery in RFC 4872.

Using protection techniques, the backup segments are pre-signaled and the forwarding entries in the backup path are already configured except for the branching points. Once the failure is detected, using for example hardware detection mechanisms or software-based techniques such as BFD (Bidirectional Forwarding Detection (BFD)), the branching points are notified and re-install the forwarding entries such that the backup paths are taken instead of the primary paths. When using dynamic re-routing techniques, the entire procedure (computing backup paths, signaling and

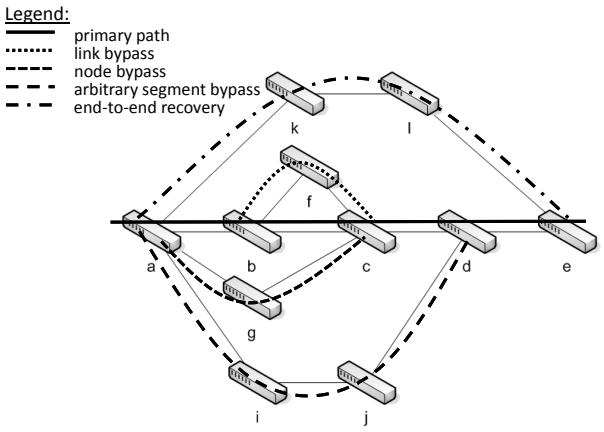


Figure 4: P2P-recovery overview

installing forwarding entries) typically happens after failure detection.

4 Point-to-multipoint (P2MP) connectivity

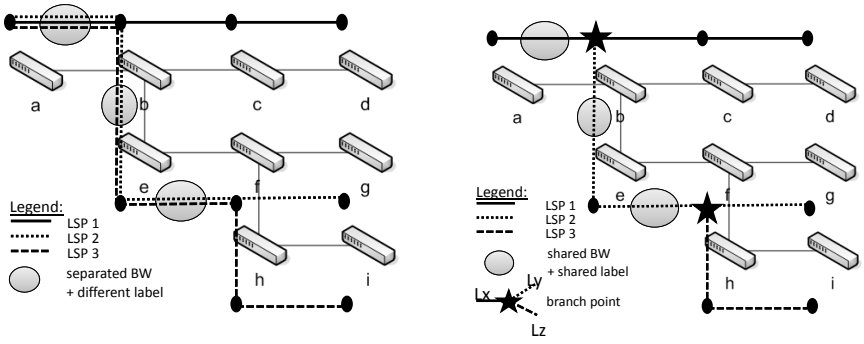
4.1 Provisioning and forwarding

A point-to-multipoint (p2mp) data path, i.e., an LSP with one source node and multiple destination nodes, can drastically increase the resource utilization. ELS allows p2mp LSPs at the logical low level forwarding structure because it is possible to link a given FTN- or ILM-entry to multiple NHLFE's. As such, an incoming frame can be replicated and sent towards multiple outgoing interfaces encapsulated with the label corresponding to the NHLFE's.

Figure 5 illustrates the gain that can be achieved by using p2mp LSPs. Part (a) of the figure, shows how source node *a* can provide three nodes, respectively *d*, *g* and *i*, with a multicast traffic stream using 3 separate p2p LSPs. It is clear from this figure that three links in the network carry more traffic volume that is strictly needed. Link a-b carries three more times and link b-e and e-f two more times the same traffic because node *a* sends the same traffic stream over the three different p2p LSPs. In addition, the mentioned links use a separate label because otherwise the intermediate nodes would not be able to distinguish between these LSPs.

The same connectivity can be achieved using p2mp LSPs with a significant gain in resource efficiency. A p2mp LSP can be constructed via incremental addition of leaves to the p2mp LSP in signaling exchange where the root is the originating switch. A p2mp LSP can be set up in three phases: i) node *a* creates an p2p LSP towards node *d*, ii) node *a* creates an p2mp LSP towards node *e* by re-using the established LSP resulting in a branch point in node *b* which then continues towards node *e*, converting the original LSP to a p2mp LSP, and iii) node a triggers the setup of additional p2mp connectivity towards node *j* by branching in point *h*.

The resulting point-to-multipoint LSP is clearly more resource efficient because: the links a-b, b-e and f-h now share the same bandwidth and label. This leads a decrease of used bandwidth and label usage of factor 3 in link a-b, and half of the used bandwidth



(a) using p2p LSPs (b) using a single multipoint LSP

Figure 5: Multipoint connectivity

in links b-e and f-h.

4.2 Recovery

Several recovery techniques exist to recover point-to-multipoint LSPs from failures. At first, all recovery techniques mentioned in Section 3 can be re-used on parts of p2mp LSPs, i.e., any segment of the p2mp (between branching points) can be protected or restored from failure given sufficient redundancy in the network (as long as primary and protecting segments are shared-risk disjoint). As such, a p2mp can be made almost ”bullet-proof” by protecting every segment of the network by a backup segment. However, this technique may quickly lead to a lot of control overhead and bad scalability properties. For example, every protected segment requires additional setup (configuration or signaling) linked to its backup segment, and every segment needs corresponding failure detection sessions over these network parts. Therefore, the goal is to find the optimal number of protected segments that optimize resource sharing and recovery time.

End-to-end recovery of p2mp LSPs can also be handled using a backup p2mp LSP. This concept is illustrated in Figure 6 where node a can provide nodes d, g and k with a multicast stream using either a primary p2mp LSP (using branch points node h and node i), or via a backup p2mp LSP (using branch points node b and f). In this example, both trees have large disjoint parts. From recoverability viewpoint this is a desirable characteristic. The setup using backup p2mp LSPs has several advantages: i) only two p2mp LSPs need to be set up (configured or signaled), and ii) multipoint failure detection sessions suffice for switch over traffic on the backup p2mp LSP. For example, multipoint BFD (see [1] and [4]) makes use of the configured p2mp LSP by allowing the source node to send Hello packets to all destination nodes in the p2mp LSP. Upon a pre-configured time-out interval (usually three times the inter-Hello time), destinations can notify the source node of a lossy p2mp LSP[1] to trigger the source node to switch over to the backup p2mp LSP. The main drawbacks of this technique are i) the notification time that is proportional to the leaf-root distance, ii) the ”freeze” of resource along the backup path (if the backup p2mp LSP is not provisioned with 0 capacity, i.e., path provisioned only), and iii) ensuring full recoverability mandates

[1] the notification usually happens via an out-of-band (OOB) control network

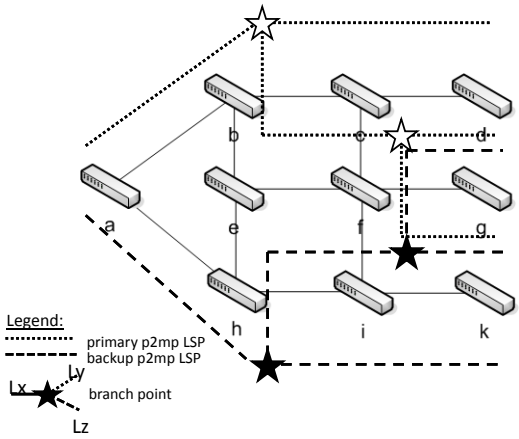


Figure 6: Backup p2mp LSP

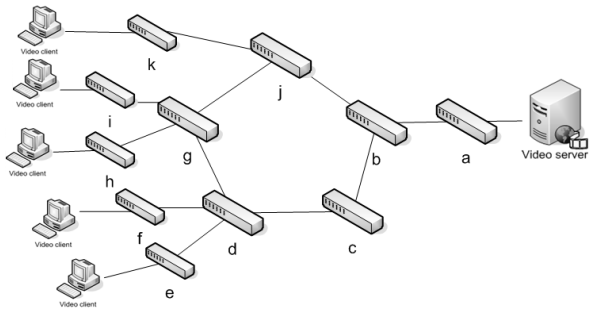


Figure 7: Network architecture

complete disjointness between the primary and the backup p2mp LSP.

5 Multipoint connectivity and recovery experimentation

In this section we will apply and evaluate the described techniques for point-to-multipoint connectivity in an emulation environment on a more realistic network scenario.

5.1 Scenario

In Figure 7 we show a small-sized network which has some similarities to a metro-access network of an Internet service provider. In the given scenario, we would like to connect a video server with a set of 5 video clients receiving the video stream being sent. As such, point-to-multipoint connectivity is desired with as much recoverability as possible, given the low redundancy in the network (only the five node ring consisting of nodes b, c, d, g, and j is able to provide a backup path).

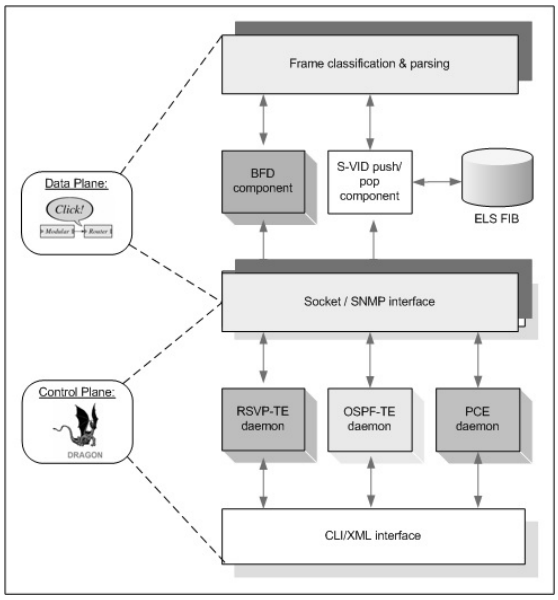


Figure 8: Node architecture in emulation platform

5.2 Emulation platform

We evaluate point-to-multipoint techniques in the emulation platform that was presented in [3]. In our emulation environment, one network device part (ELS switch) is imitated by a computation system running custom software, e.g., a server blade or PC. Therefore, to emulate a whole network setup, typically a set of server blades or PCs is needed (a set of Linux PC's).

For benchmarking ELS technology, an execution environment was set up using the Click Modular Router that emulates the forwarding plane ([2]), and the Dragon VLSR GMPLS software that emulates the control plane[2] ([5]). The resulting node architecture is shown in Figure 8.

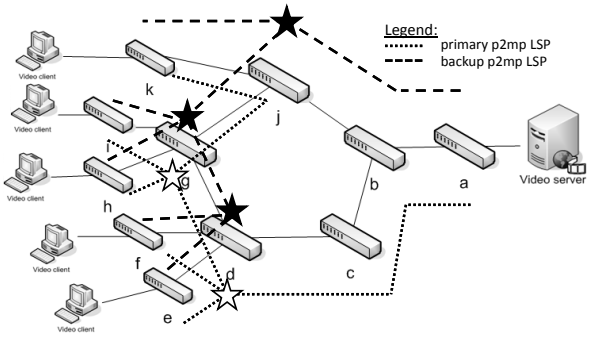
To allow arbitrary topologies with ELS emulation software, Emulab software was used on our local virtual wall at IBBT. Using a ns2 look-alike configuration script, Emulab software allows to define which software image needs to be installed on which PC part of the execution environment, and how several PC's need to be interconnected to each other, defining thus the topology.

5.3 Recovery of p2mp connectivity

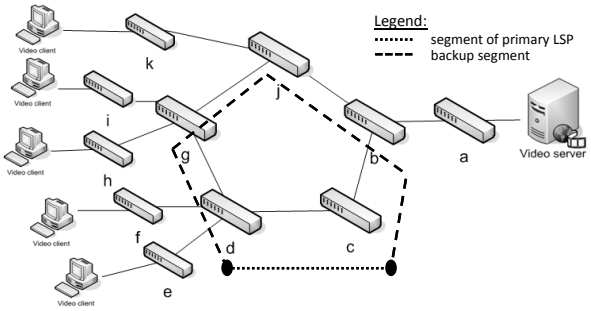
For the given scenario, it is not difficult to set up a p2mp LSP in order to fulfill the requirement of providing all video clients with the video stream generated at source node a. We chose primary p2mp LSP shown in part (a) of Figure 9 as basis.

Next, we evaluated two variants of p2mp LSP recovery: i) protecting the p2mp LSP using a backup p2mp LSP, and ii) recovering the p2mp LSP by means of a collection of protected segments. For the first, the chosen backup tree is shown in 9 part (a), for the latter, we have protected all links of the ring composed by nodes b, c, d, g and j by the corresponding other half of the ring. The segment protection concept of the latter

[2]Dragon was only used for signaling p2p LSPs, p2mp LSPs were configured by management software



(a) primary and backup p2mp LSP



(b) segment protection in p2mp LSP

Figure 9: Recovery of p2mp LSPs

is shown for the protection of link d-c using the backup segment composed of the links d-g, g-j, j-b and b-c.

Failure detection is provided by our BFD implementation in Click. The implementation allows to detect failures at the following levels: i) link-failure detection, end-to-end failure detection and multipoint-failure detection (see [3] and [4]).

- For the first recovery variant, we only needed to set up the primary and backup LSP, as well as two multipoint BFD sessions. When connectivity is lost with the source node in one of the video clients, a notification message was sent using a out-of-bound (OOB) control network[3] to the source (see Figure 10), triggering switch-over to the backup LSP (or vice versa for the switch-back).

- For the second recovery variant, the setup of 5 backup segments as well as 5 BFD sessions were needed to ensure the protection of the links in the middle ring of the network. When some of the nodes in the ring b-c-d-g-j detects that a link has failed, traffic is sent via the backup segment towards the next hop. Figure 11 shows the recovery process when node c has detected that link d-c fails. From that moment incoming traffic at node c is sent to node d via the links c-b, b-j, j-g and g-d. The original p2mp LSP can now be continued in node d.

[3]consisting of a single switch connecting all nodes

Table 1: Recovery time upon failing link b-c or c-d

Timeline for variant 1:	Timeline for variant 2:	Event:
0	0	Failure
25	25	BFD Failure notification (5x5)
26	26	Trigger protection switch.
27	27	Send notify to branch/merge
35≤x≤38	35	Receive notify ack
35≤x≤38	35	Start fast-switchover
38≤x≤41	38	Switch-over finished

The recovery time needed for both approaches was very similar as it can be observed in Table 1 when a link b-c or c-d was failing. As it can be seen in this table, the main difference results from the time needed for notifying the branch point. Whereas for the first variant (backup p2mp LSP), a notification needs to be sent to the source node a upon failure detection, the variant based on segment recovery could almost[4] directly trigger the switch-over of traffic. However, in our setup the observed difference was very minimal, because the largest time went to the internal control processing.

In Figure 12 part (a), the performance of our BFD implementation is shown. It can be observed, depending on the length of paths (in hops) being monitored influence the lowest failure detection performance we could achieve. Up to 18 nodes, we were able to reach detection times of 15 ms using Hello-transmission speeds (TX) and equal reception timeouts (RX) of 5 ms (a failure being reported upon 3 missing Hello's). Part (b) of the figure, illustrates that the gap in the throughput using the above techniques is negligible compared to legacy bridged Ethernet failure detection and re-convergence based on RSTP.

Given these results, one could conclude that the there is no real reason to go for the more resource consuming variant which sets up recovery segments for all links. However, as can be seen from Figure 9 part (a) and the described mechanism, the corresponding recovery mechanism is not able to fully recover from a failure in link d-g. This is the case, because the link is both needed by the primary and backup p2mp LSP. This illustrates that, depending on the specific network topology and likelihood of failures in specific links, that setting up dedicated segment protection can be worth to evaluate.

6 Conclusion

This paper introduced Ethernet Label Switching (ELS) as an efficient carrier-grade Ethernet technology for the provision and recovery of point-to-multipoint connectivity. Several recovery mechanisms were discussed and benchmarked in an emulation environment on a realistic network scenario. We showed that several alternatives are able to provide high recoverability, however some trade-offs must be considered depending on the specific failure likelihood and network topology.

[4]depending on the endpoint of the link detecting the failure first

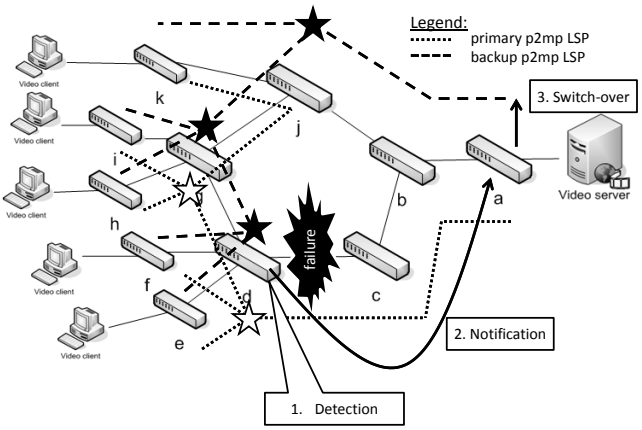


Figure 10: Recovery process with backup p2mp LSP

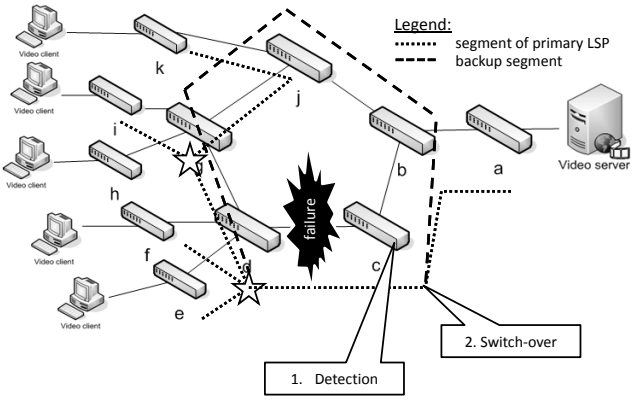
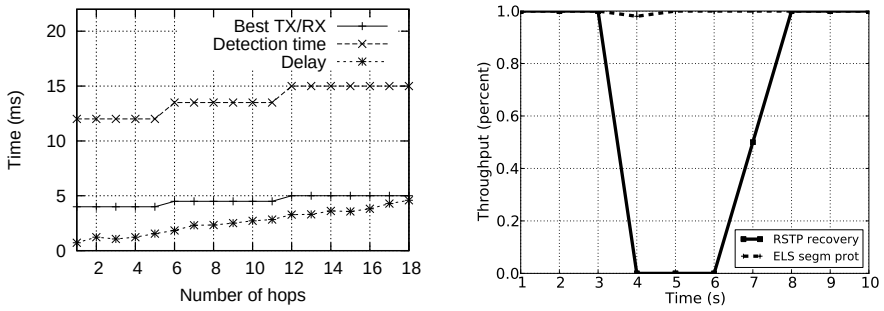


Figure 11: Recovery process using segment recovery



(a) BFD time performance

(b) Throughput during recovery

Figure 12: Performance of failure detection and recovery operation

7 Acknowledgment

This research is partly funded by The Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen) through the TIGER project in the European CELTIC framework and the FP-7 project BONE.

References

[1] D. Katz and D. Ward. BFD for Multipoint Networks. Internet-Draft draft-katz-ward-bfd-multipoint-02, Internet Engineering Task Force, February 2009. Work in progress.

[2] Robert Morris, Eddie Kohler, John Jannotti, and M. Frans Kaashoek. The click modular router. In *SOSP '99: Proceedings of the seventeenth ACM symposium on Operating systems principles*, pages 217–231, New York, NY, USA, 1999. ACM.

[3] Wouter Tavernier, Dimitri Papadimitriou, Didier Colle, Mario Pickavet, and Piet Demeester. Emulation of gmpls-controlled ethernet label switching. *Testbeds and Research Infrastructures for the Development of Networks and Communities, International Conference on*, 0:1–9, 2009.

[4] Wouter Tavernier, Dimitri Papadimitriou, Bart Puype, Didier Colle, Mario Pickavet, and Piet Demeester. Fast failure detection in multipoint networks. In Giorgio Nunzi, Caterina M. Scoglio, and Xing Li, editors, *IPOM*, volume 5843 of *Lecture Notes in Computer Science*, pages 51–64. Springer, 2009.

[5] Xi Yang, Chris Tracy, Jerry Sobieski, and Tom Lehman. Gmpls-based dynamic provisioning and traffic engineering of high-capacity ethernet circuits in hybrid optical/packet networks. In *INFOCOM*, 2006.